Amendments to the Claims:

This listing of claims is to replace all prior versions and listings of claims in this application:

1. (Currently Amended) A method comprising:

receiving an original digital good; ~~and~~

randomly applying various forms of protection to a plurality of segments of the original digital good to generate a plurality of protected segments to be included in a protected digital good;

generating a plurality of checkpoints, each of the checkpoints being associated with at least one of the protected segments, the checkpoint being operable to cause a system receiving the protected digital good to validate that at least one protected segment with which the checkpoint is associated has not been tampered with; and

assembling the protected digital good by collecting the plurality of protected segments, wherein at least two of the segments overlap one another, the overlapping segments being different from each other, and the checkpoints are inserted in the protected digital good at varying positions relative to the protected segments with the checkpoints are associated.

2. (Original) A method as recited in claim 1, wherein the randomly applying comprises pseudo randomly applying the various forms of protection according to pseudo random techniques.

3. (Original) A method as recited in claim 1, wherein the applying comprises randomly selecting the forms of protection from a set of available forms of protection.

4. (Original) A method as recited in claim 1, wherein the applying comprises applying the various forms of protection to randomly selected portions of the original digital good.

5. (Original) A method as recited in claim 1, wherein the various forms of protection are selected from a group of protection tools comprising code integrity verification, acyclic code integrity verification, cyclic code integrity verification, secret key scattering, obfuscated function execution, encryption/decryption, probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo random number generators with time varying inputs, anti-disassembly methods, varying execution paths between runs, anti-debugging methods, and time/space separation between tamper detection and response.

6. (Original) A method as recited in claim 1, wherein the applying comprises applying a form of protection in which a checksum can be computed on a set of bytes of the digital good without actually reading the bytes.

7. (Original) A computer-readable medium comprising computer-readable instructions that, when executed by a processor, direct a computer system to perform the method as recited in claim 1.

8. (Currently Amended) A method comprising:

segmenting a digital good into a plurality of segments;

selecting multiple segments from the plurality of segments; ~~and~~

transforming only the selected segments according to different protection techniques to produce a protected digital good having a composite of variously protected segments;

augmenting at least one segment using a certain protection technique; and

inserting a checkpoint within the protected digital good but outside of the augmented segment and at a varying position relative to the augmented segment, the checkpoint being suitable to evaluate a validity of the augmented segment.

9. (Original) A method as recited in claim 8, wherein at least two of the segments overlap one another.

10. (Original) A method as recited in claim 8, wherein the selecting comprises randomly selecting the segments.

11. (Original) A method as recited in claim 8, wherein the transforming comprises transforming the selected segments according to randomly chosen protection techniques.

12. (Canceled).

13. (Original) A method as recited in claim 8, further comprising receiving quantitative parameters indicative of how much the protected digital good should be altered.

14. (Original) A method as recited in claim 13, wherein the transforming is performed to satisfy the quantitative parameters.

15. (Original) A method as recited in claim 8, wherein the protection techniques are selected from a group of protection tools comprising code integrity verification, acyclic code integrity verification, cyclic code integrity verification, secret key scattering, obfuscated function execution, encryption/decryption, probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo random number generators with time varying inputs, anti-disassembly methods, varying execution paths between runs, anti-debugging methods, and time/space separation between tamper detection and response.

16. (Original) A method as recited in claim 8, wherein the transforming comprises applying a protection technique in which a checksum can be computed on a set of bytes of the digital good without actually reading the bytes.

17. (Original) A computer-readable medium comprising computer-readable instructions that, when executed by a processor, direct a computer system to perform the method as recited in claim 8.

18. (Currently Amended) A method comprising:

establishing parameters prescribing a desired quantity of protection to be applied to a software product in generating a protected software product;

parsing the software product into code sections;

selecting at least one code section;

augmenting the selected code section to add protection qualities to generate an augmented code section;

generating a checkpoint configured to cause a system receiving the augmented code section to attempt to validate that the augmented code section has not been tampered with;

determining a checkpoint position for the checkpoint to be inserted in the protected software product, the checkpoint position being outside of a position of the augmented code section and at an offset to the augmented section that is varied from additional checkpoint positions associated with other augmented code sections; and

repeating the selecting and the augmenting for different code sections until the desired quantity of protection has been applied.

19. (Original) A method as recited in claim 18, wherein the establishing comprises enabling a user to enter the parameters.

20. (Original) A method as recited in claim 18, wherein the augmenting comprises applying a protection technique selected from a group of protection techniques comprising code integrity verification, acyclic code integrity verification, cyclic code integrity verification, secret key scattering, obfuscated function execution, encryption/decryption, probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo random number generators with time varying inputs, anti-disassembly methods, varying execution paths between runs, anti-debugging methods, and time/space separation between tamper detection and response.

21. (Original) A method as recited in claim 18, wherein the augmenting comprises applying a protection technique in which a checksum can be computed on a set of bytes of the digital good without actually reading the bytes.

22. (Original) A computer-readable medium comprising computer-readable instructions that, when executed by a processor, direct a computer system to perform the method as recited in claim 18.

23. (Currently Amended) A production system, comprising:

a memory to store an original digital good; ~~and~~

a production server equipped with a set of multiple protection tools that may be used to augment the original digital good for protection purposes, the production server being configured to:

parse the original digital good and apply protection tools selected from the set of protection tools only to selected portions of the original digital good in a random manner to produce a protected digital good having a composite of the protected selected portions;

generate a plurality of checkpoints, each of the checkpoints being associated with one of the protected selected portions and causing a system receiving the protected digital good, upon encountering each of the checkpoints, to attempt to validate the protected selected portions associated with each of the checkpoints; and

insert the plurality of checkpoints within the protected digital good, the positions of each of the plurality of checkpoints being one of variably offset and randomly offset outside of the protected selected portions with which each of the checkpoints is associated.

24. (Original) A production system as recited in claim 23, wherein the protection tools are selected from a group of protection tools comprising code integrity verification, acyclic code integrity verification, cyclic code integrity verification, secret key scattering, obfuscated function execution, encryption/decryption, probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo random number generators with time varying inputs, anti-disassembly methods, varying execution paths between runs, anti-debugging methods, and time/space separation between tamper detection and response.

25. (Original) A production system as recited in claim 23, wherein the production server applies a protection tool that enables a checksum to be computed on a set of bytes of the digital good without actually reading the bytes.

26. (Original) A production system as recited in claim 23, wherein the production server has a pseudo random generator to introduce randomness into the application of the protection tools to various portions of the original digital good.

27. (Currently Amended) An obfuscation system, comprising:

a parser to parse a digital good into a plurality of segments;

a set of protection tools that may be applied to the segments of the digital good to augment the segments with protection qualities;

a target segment selector to select at least one segment from the plurality of segments; ~~and~~

a tool selector to select at least one protection tool from the set of protection tools and apply the selected protection tool to the selected segment so that a protection tool of the set of protection tools is applied only to a selected segment of the plurality of segments to generate a plurality of protected selected segments; and

a checkpoint generator to create checkpoints for at least a portion of the protected selected segments, the checkpoints being assigned positions outside of the protected selected segments at variable positions relative to each of the protected selected segments, the checkpoints being operable to cause a system receiving the plurality of protected selected segments to attempt to validate authenticity of the protected selected segments.

28. (Original) An obfuscation system as recited in claim 27, wherein the protection tools are selected from a group of protection tools comprising code integrity verification, acyclic code integrity verification, cyclic code integrity verification, secret key scattering, obfuscated function execution, encryption/decryption, probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo random number generators with time varying inputs, anti-disassembly methods, varying execution paths between runs, anti-debugging methods, and time/space separation between tamper detection and response.

29. (Original) An obfuscation system as recited in claim 27, wherein the target segment selector comprises a pseudo random generator to enable random selection of the segment.

30. (Original) An obfuscation system as recited in claim 27, wherein the tool selector comprises a pseudo random generator to enable random selection of the protection tool.

31. (Original) An obfuscation system as recited in claim 27, further comprising a quantitative unit to specify a quantity of protection qualities to be added to the digital good.

32. (Currently Amended) A client-server system, comprising:

a production server to randomly apply various forms of protection only to selected portions of a digital good to produce a protected digital good, the protected digital good including a plurality of one of variably and randomly placed checkpoints configured to cause a system encountering the checkpoints to attempt to authenticate that the selected portions of the protected digital good have not been tampered with; and

a client to store and execute the protected digital good, the client being configured to, upon encountering each of the checkpoints, to evaluate the selected portions of the protected digital good to determine whether the protected digital good has been tampered with.

33. (Currently Amended) One or more computer-readable media having computer-executable instructions that, when executed, direct a computing device to:

parse a digital good into a plurality of segments; ~~and~~

apply multiple different protection tools to only a selected portion of the segments in a random manner to produce a protected digital good having a composite of variously protected portions; and

insert a plurality of checkpoints into the protected digital good at positions one of variably and randomly offset from the variously protected portions, such that upon encountering each of the plurality of checkpoints, a receiving computing system attempting to execute the protected digital good will to authenticate that variously protected portions have not been tampered with.

34. (Original) One or more computer-readable media as recited in claim 33, further comprising computer-executable instructions to randomly select the protection tools from a set of available protection tools.

35. (Original) One or more computer-readable media as recited in claim 33, further comprising computer-executable instructions to apply the protection tools to randomly selected portions of the original digital good.

36. (Original) One or more computer-readable media as recited in claim 33, wherein the protection tools are selected from a group of protection tools comprising code integrity verification, acyclic code integrity verification, cyclic code integrity verification, secret key scattering, obfuscated function execution, encryption/decryption, probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo random number generators with time varying inputs, anti-disassembly methods, varying execution paths between runs, anti-debugging methods, and time/space separation between tamper detection and response.